

- 1) Local
- 2) instance
- 3) static

class A

```
int a=10;           → instance
static int b=20;    → static variable
void add()          → local
int c=30, d;
d=a+b+c; ✓
S.o.p(d);
```

Void null()

```
int e=40, f;
f=aXbx
Sop(e);   e; //
```

class A

{

int a=10;
static int b=20;

void add()

{

int c=30,d;
d=a+b+c;

Sop(d);

3

void mul()

{

int e=40,f;
f=a*b*c; //error
Sop(f);

3

LOCAL VARIABLES

1) DECLARATION:- inside methods, constructors or blocks.

2) SCOPE:- inside method, constructor or block not outside.

3) WHEN VARIABLES - when method, constructor or block GETS ALLOCATED - gets executed . variables allocates memory
- when gets exits, variables destroyed

4) STORED MEMORY:- Stack memory

5) DEFAULT VALUES - doesn't have any default values,
value should be provided before use

6) ACCESS SPECIFIERS:- cannot be used with local variables

Scope Of Instance Variable :

inside all methods, blocks and constructors within a class (but not inside the static method directly)
(to use within static method we have to create an object of the class then by object reference we have to call the variable)

Here is an example :

Abc - Notepad

```
File Edit Format View Help
class Abc
{
    int a=10;           instance variable
    static int b=20;     static variable
    public static void main(String[] args)
    {
        Abc ob1=new Abc();

        System.out.println(a); //error
        System.out.println(ob1.a); correct
        System.out.println(b);
        System.out.println(ob1.b); output will be :
    }
}
```

Command Prompt

```
D:\>javac Abc.java
Abc.java:9: error: non-static variable a cannot be referenced from a static context
    System.out.println(a); //error
                                         ^
1 error
D:\>
```

Default Values Of Instance Variable :

They have default values like int is 0, boolean is false, String is null etc.

if we directly print instance variable it will print default values but if we print local variable directly it will provide error

Here is an example :

```
Abc - Notepad
File Edit Format View Help
public class Abc
{
    int a;—— instance variable
    String s;——instance variable
    public static void main(String[] args)
    {
        int b;—— local variable

        Abc ob=new Abc();
        System.out.println("value of a : "+ob.a);
        System.out.println("value of s : "+ob.s);
        System.out.println("value of b : "+b);
    }
}
```

(correct)
output will be:
0
null

```
D:\>javac Abc.java
Abc.java:12: error: variable b might not have been initialized
    System.out.println("value of b : "+b);
                           ^
1 error
D:\>
```

INSTANCE VARIABLES

```

class A
{
    int a=10;           instance
    static int b=20;
    void add()
    {
        int c=30, d;   local
        d=a+b+c;
        Sop(d);
    }
    void mul()
    {
        int e=40, f;   local
        f=a*b*c;
        Sop(f);
    }
}
  
```

- 1) DECLARATION:- inside the class but outside methods, constructors or blocks.
- 2) SCOPE:- inside all methods, blocks & constructors within a class (not inside the static method directly)
- 3) WHEN VARIABLES GETS ALLOCATED:- when object is created variables allocated; Object gets destroyed variable released memory.
- 4) STORED MEMORY:- Heap-memory
- 5) DEFAULT VALUES:- They have default values like int → 0, boolean → false, object → null
- 6) ACCESS SPECIFIERS:- can be used.
- 7) How to access:- it can be called directly

static method - A ob=new A();
ob.a;

STATIC VARIABLES

```
class A
{
    int a=10;           instance
    public static int b=20;
    void add()
    {
        int c=30;      local
        d=a+b+c;
        Sop(d);
    }
    static void mul()
    {
        int e=40;      local
        f=axboxexc;
        Sop(f);
    }
}
```

- 1) DECLARATION:- with static keyword in a class but outside a method, constructor or block
- 2) SCOPE:- similar to instance variable ie. inside a method, constructors or blocks including static " "
- 3) WHEN VARIABLES GETS ALLOCATED:-
 - when we run program & class file is loaded, variable allocated
 - when class file gets unloaded, variable gets destroyed.
- 4) STORED MEMORY:- non-heap memory or static memory
- 5) DEFAULT VALUES:- Similar to instance variables
- 6) Access SPECIFIERS:- can be used
- 7) How to access:-
 - 1) directly
 - 2) by using class name (A.b)
 - 3) by using object ref. name

```
class Abc
```

{

```
    int a=10;
```

```
    static int b=20;
```

```
    public static void main (String args[])
```

{

```
        Abc ob1=new Abc();
```

```
        Sop(ob1.a); // 10  
        Sop(ob1.b); // 20
```

```
        ob1.a=1000;
```

```
        ob1.b=2000;
```

```
        Sop(ob1.a); // 1000  
        Sop(ob1.b); // 2000
```

```
        Abc ob2=new Abc();
```

```
        Sop(ob2.a); // 10  
        Sop(ob2.b); // 2000
```

